

# Convolutional Imputation of Matrix Networks

Qingyun Sun<sup>1</sup> Mengyuan Yan<sup>2</sup> Stephen Boyd<sup>2</sup>

## Abstract

A matrix network is a family of matrices, where each node represents a matrix and the relation between them is modeled as a weighted graph. We consider a novel sampling scheme where a fraction of matrices might be completely unobserved. How can we recover the entire matrix network from incomplete observations? This mathematical problem arises in many applications including medical imaging with anisotropical sampling and social network.

To recover the matrix network, we propose a structural assumption that the matrix network can be approximated by the *generalized convolution* of low rank matrices living on the same network. We propose an iterative imputation algorithm to complete the matrix network. We characterize the exact recovery regime for varying rank and average sampling rate both theoretically and numerically, and we discover a phase transition phenomenon that is universal with different graph structures.

## 1. Introduction

In machine learning and social network problems, information is often encoded in matrix form. User profiles in social networks can be embedded into feature matrices; item profiles in recommendation systems can also be modeled as matrices. Many medical imaging modalities, such as MRI and CT, also represent data as a stack of images. These matrices have underlying connections that can come from spatial or temporal proximity, or observed similarities between the items being described, etc. A weighted graph can be built to represent the connections between matrices. Therefore we propose matrix networks as a general model

<sup>1</sup>Department of Mathematics, Stanford University, Stanford, CA, USA <sup>2</sup>Department of Electrical Engineering, Stanford University, Stanford, CA, USA. Correspondence to: Stephen Boyd <boyd@stanford.edu>.

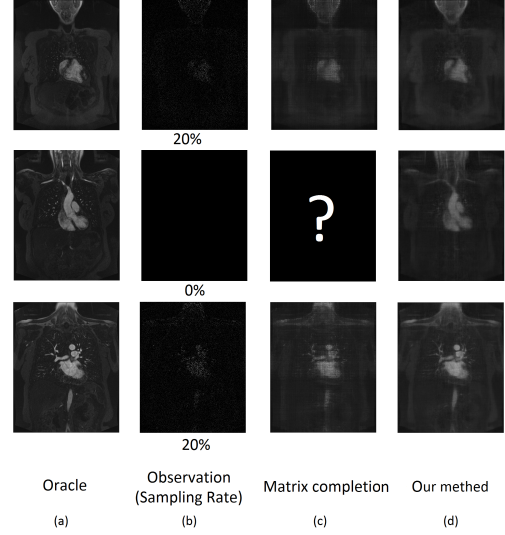


Figure 1. (a) Original MRI image frames (oracle). (b) Our sampled and corrupted observation. (c) Recovered image frames using nuclear norm minimization on individual frames. (d) Recovered image frames using our convolutional imputation algorithm.

for data representation. A matrix network is defined by a weighted undirected graph whose nodes are matrices.

Due to the limitations of data acquisition processes, sometimes we can only observe a subset of entries from each data matrix. The fraction of entries we observe may vary from matrix to matrix. In many real problems a subset of matrices can be completely unobserved, leaving no information for ordinary matrix completion methods to recover the missing matrices. To our knowledge we are the first to examine this novel sampling scheme. Since we model the matrices as nodes on a graph, information from neighboring matrices makes it possible to recover the missing ones.

As an example, in the following MRI image sequence (Figure 1(a)-(c)), we observe noisy entries of each frame of the MRI images. Two frames out of a total of 88 are completely unobserved, and the other 86 frames are sampled with 20% sampling rate using an i.i.d. Bernoulli distribution. All observed entries are corrupted by independent Gaussian noise. If we perform matrix completion by nu-

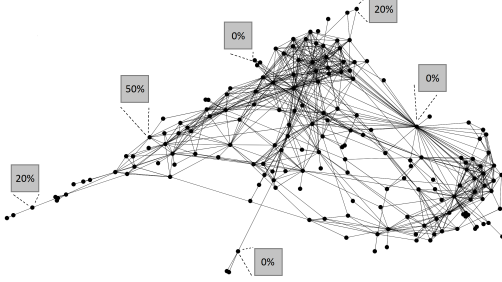


Figure 2. An example of matrix network. Each node on the graph represents a matrix. Our observations are subsets of entries from these matrices. When some matrices are fully unobserved, information from its neighborhood is needed for the recovery of missing ones.

clear norm minimization on individual frames, we are not able to recover the completely unobserved matrices. When we build a network on the image frames, in this case a  $1-D$  chain representing the sequence, we are able to recover the missing frames by borrowing information from neighboring frames, as shown in Figure 1(d).

The ability to recover all matrices from partial observations, especially inferring matrices that are totally unobserved, is crucial to many applications such as the cold start problem in networks. Illustrated in Figure 2, new items or users in a network, which does not have much information available, need to aggregate information from the network to have an initial estimate of their feature matrices, in order to support inference and decisions.

We make the following major contributions in this paper:

1. We proposed the matrix network model, which encodes both the matrices and their relations. We demonstrate that this model is highly beneficial for completion tasks.
2. We defined generalized convolution on matrix networks, and we proposed the structural assumption that a matrix network is the generalized convolution of two sets of low rank matrices living on the same network.
3. We propose a convolutional imputation algorithm to complete the matrix network from partial observations where many matrices that are completely unobserved.
4. We prove exact recovery guarantee when the average sampling rate is  $O(\frac{r}{n} \log^2(nN))$ . Furthermore, we numerically characterize the precise success regime for rank and the average sampling rate. We discover a phase transition phenomenon that is universal in different graph structures.

This paper is organized as follows: After a brief review of related work in section 2, we will give the mathematical definition of a matrix network and of generalized convo-

lution in section 3. In section 4, we give the formal formulation of the completion problem for matrix networks, and propose the structural assumption that the matrix network can be approximated by generalized convolution of low rank matrix networks. Based on the statistical model we propose an iterative imputation algorithm in section 5. In section 6, we provide theoretical analysis for exact recovery guarantee, and numerical study on the phase transition phenomenon. More experimental results are in section 7 and we conclude in section 8.

## 2. Related work

Low-rank matrix recovery is an important field of research. Hastie *et al.* (Hastie & Mazumder, 2015; Hastie *et al.*, 2015; Mazumder *et al.*, 2010) proposed the soft-impute algorithm, as an iterative method to solve large scale matrix completion problem. The soft-impute algorithm inspired our imputation algorithm. There were also a long line of works building theoretical tools to analyze the recovery guarantee for matrix completion (Candès & Recht, 2012; Cai *et al.*, 2010; Candès & Tao, 2010; Candès & Plan, 2010; Donoho *et al.*, 2014; Gavish & Donoho, 2014; Keshavan *et al.*, 2010b;a). Besides matrix completion, Gross analyzed the problem of efficiently recovering a low-rank matrix from a fraction of observations in any basis (Gross, 2011). These works enlightened our exact recovery analysis.

Tensor could be considered as matrix network when the network is one-dimensional lattice. Several works on tensor completion defined nuclear norm for tensors as linear combinations of the nuclear norm of its unfoldings (Gandy *et al.*, 2011; Liu *et al.*, 2013; Filipović & Jukić, 2015). Besides the common CP and Tucker decompositions of tensor, the recent work (Zhang & Aeron, 2016; Kernfeld *et al.*, 2015; Liu *et al.*, 2016) defined t-product using convolution operators between tensor fibers, which was close to the convolution of matrix network using discrete Fourier matrix, and they applied the method in indoor localization. Different from previous works, we considered a new sampling scheme that some matrices was completely unobserved, and undersampling ratio could be highly unbalanced for the other observed matrices. This sampling scheme was not considered before, yet it was very natural under the matrix network model.

Low-rank matrix recovery could be viewed as a non-commutative analogue of compressed sensing by replacing sparse vector with low-rank matrix. In compressed sensing, the setting with block diagonal sensing matrix was studied by the recent work (Monajemi & Donoho, 2017). They studied the recovery of sparse tensor from anisotropic sampling scheme, and demonstrated a phase transition that was different from the well-known phase transition for classical

Gaussian/Fourier sensing matrix and isotropic sampling. In our low rank matrices recovery problem, our novel sampling scheme also corresponded to a block diagonal operator. We demonstrated a new phase transition phenomenon that was universal for different graph structures.

The idea of graph Fourier transform was rooted in spectral graph theory. Recent advances in the field of convolutional neural network by Yann LeCun *et al.* (Bruna et al., 2013; Henaff et al., 2015) extended the neural network from Euclidean grids to graphs. Our work proposed the framework of matrix networks, which could be used to model low rank structures in weight matrices on graph convolutional neural network.

### 3. Mathematical definitions

**Matrix network** First, let's consider any weighted graph  $G$ , which has  $N$  nodes and an adjacent matrix  $W \in \mathbf{R}^{N \times N}$ , where  $W_{ij}$  is the weight on the edge between node  $i$  and  $j$ . The nodes are indexed  $1, 2, \dots, N$ , the index set is called  $J = \{1, 2, \dots, N\}$ . In the following we use the index set  $J$  to represent the set of corresponding nodes on the graph.

We define a matrix network by augmenting this weighted graph  $G$  with a matrix valued function  $A$  on the set of nodes  $J$  in the graph  $G$ . The function  $A$  maps each node in the graph to a complex valued matrix of size  $m \times n$ , denoted as  $A : J \rightarrow \mathbf{C}^{m \times n}$ . For any node index  $i$ ,  $A(i)$  is the complex valued matrix at node  $i$ . Figure 3 illustrates an example of a matrix network with three nodes on a clique. For convenience, we also denote the matrix network as  $A$ , which refers to the weighted graph and the matrix valued function  $A$  on this graph.

We define a  $L_2$  norm  $\|\cdot\|$  on the matrix network  $A$  by the squared sum of all entries in all matrices of the network. We can equivalently define it as the sum of Frobenius norm  $\|\cdot\|_F$  of all matrices, denoted as

$$\|A\|^2 = \sum_{i=1}^N \|A(i)\|_F^2 = \sum_{i=1}^N \sum_{a=1}^m \sum_{b=1}^n (A(i)_{ab})^2.$$

**Graph Fourier transform** Graph Fourier transform is an analog of the Discrete Fourier Transform. For a weighted undirected graph  $G$  and its adjacent matrix  $W$ , the normalized graph Laplacian is defined as  $L = I - D^{-1/2} W D^{-1/2}$ , where  $D$  is a diagonal matrix with entries  $D_{ii} = \sum_j W_{ij}$ . The graph Fourier transformation matrix  $U$  is defined as

$$UL = U\Lambda,$$

where  $\Lambda$  is the diagonal matrix of the eigenvalues of  $L$ . Therefore,  $U$  is an unitary  $N \times N$  matrix, and the eigenvectors of  $L$  are the row vectors of  $U$ . The graph Fourier transform of a scalar valued function on graph  $f : J \rightarrow \mathbf{R}$ ,

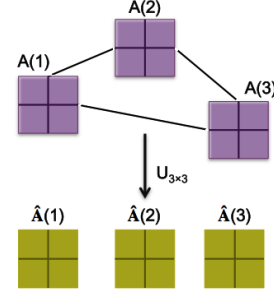


Figure 3. Demonstration of graph Fourier transformation on matrix network.

is  $F_k = \sum_i U_{k,i} f_i$ , and  $F$  is a scalar function defined only on the eigenvalues of  $L$ , here we rank the eigenvalues in descending order and identify the  $k$ -th eigenvalues with its index  $k$  for simplicity.

We remark that discrete Fourier transform is one special example of graph Fourier transform. When the graph is an one dimensional periodic grid of period  $N$ ,  $L$  is the discrete Laplacian matrix, and the eigenvectors are just the basis vectors for discrete Fourier transform, which are sine and cosine functions with different frequencies.

**Graph Fourier transform of matrix networks** Now we define the graph Fourier transform of matrix networks. For a matrix network  $A$ , we define its graph Fourier transform  $\hat{A} = UA$ , as a stack of  $N$  matrices in the spectral space of the graph. Each matrix is a linear combination of matrices on the graph, weighted by the graph Fourier basis.

$$\hat{A}(k) = \sum_{i \in J} U(k, i) A(i).$$

Intuitively, if we view the matrix network  $A$  as a set of  $m \times n$  scalar functions on the graph, the graph Fourier transform on matrix network is applying graph Fourier transform on each function individually. Using tensor notation, the element of  $A$  is  $A(i, a, b)$ ,  $i = 1, \dots, N$ ,  $a = 1, \dots, m$ ,  $b = 1, \dots, n$ , the graph Fourier transform  $U$  can be represented by a big block diagonal matrix  $U \otimes I$  where each block is  $U$  of size  $N^2$ , and there are  $(mn)^2$  such blocks.

The inverse transform  $U^{-1} = U^*$  will transform  $\hat{A}$  back to  $A$ , as

$$A(i) = \sum_{k=1}^N U^*(i, k) \hat{A}(k).$$

**Generalized convolution of matrix networks** We can extend the definition of convolution to matrix networks. For two matrix networks  $X : J \rightarrow \mathbf{C}^{m \times r}$  and  $Y : J \rightarrow$

$\mathbb{C}^{r \times n}$  on the same graph, we define their generalized convolution by generalizing the convolution theorem with respect to Fourier transform:

$$(\widehat{X \star Y})(k) = \hat{X}(k)\hat{Y}(k).$$

Then  $\widehat{X \star Y}$  is a stack of matrix where each matrix is the matrix multiplication of  $\hat{X}(k)$  and  $\hat{Y}(k)$ .

#### 4. Completion problem with missing matrices

Imagine that we observe a few entries  $\Omega(i)$  of each matrix  $A(i)$ . We define the sampling rates as  $p_i = |\Omega(i)|/(mn)$ . The projection operator  $P_\Omega$  is defined to project the full matrix network to our partial observation by only retaining entries in the set  $\Omega = \bigcup \Omega(i)$ .

The sampling rate can vary from matrix to matrix. The main novel sampling scheme we include here is that a subset of matrices may be completely unobserved, namely  $p_i = 0$ . This sampling scheme almost has not being discussed in depth in the literature. Most approaches in matrix and tensor completion fail under this sampling scheme. The difficulty lies in the fact if a matrix is fully unobserved, there is no information at all from itself for the recovery, therefore we must leverage the information from other observed matrices.

To focus on understanding the essence of this difficulty, it is worth considering the extreme sampling scheme where each matrix is either fully observed or fully missing, which we call **node undersampling**.

To recover missing entries, we need structural assumptions about the matrix network  $A$ . We propose the assumption that  $A$  can be well-approximated by the generalized convolution  $X \star Y$  of two matrix networks  $X, Y$  of size  $m \times r$  and  $r \times n$ , for some  $r$  much smaller than  $m$  and  $n$ . We will show that under this assumption, accurate completion is possible even if a significant fraction of the matrices are completely unobserved.

We verified the above assumption in an MRI scan example. The scan has a stack of 88 images of dimension  $512 \times 416$ . We modeled the underlying graph as a chain, and calculated the graph Fourier transform of the images. The resulting matrices in the spectral space are low rank, as shown in Figure 4.

We formally formulate the completion problem as follows. Let  $A^0 = X^0 \star Y^0$  be a matrix network of size  $m \times n$ , as the ground truth, where  $X^0$  and  $Y^0$  are matrices of size  $m \times r$  and  $r \times n$  on the same network. After the graph Fourier transform,  $\hat{A}^0(k)$  are rank  $r$  matrices.

We first consider the noiseless setting where our observation is  $A^\Omega = P_\Omega(A^0)$ . In this case, we can consider the following nuclear norm minimization problem, as a convex

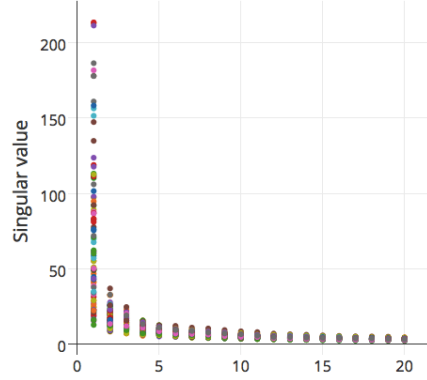


Figure 4. Spectral space singular value distribution of a MRI scan. Here a MRI scan consisting of 88 images with dimension  $512 \times 416$  is transformed into spectral space, and the singular values for each spectral space matrix is plotted in descending order. Only the first 20 largest singular values are shown for each matrix, and each color represent singular values from one matrix in spectral space. For all the spectral space matrices, singular values quickly decrease to almost zero, demonstrating that they are in fact low rank.

relaxation of rank minimization problem,

$$\begin{aligned} & \underset{\hat{M}}{\text{minimize}} \quad \sum_{k=1}^N \|\hat{M}(k)\|_*, \\ & \text{subject to} \quad A^\Omega = P_\Omega \mathcal{U}^* \hat{M}. \end{aligned}$$

As extension, we can include the noisy observation setting: Suppose our observations satisfy the Gaussian noise model  $P_\Omega(A) = P_\Omega(A^0) + P_\Omega(W)$ , each entry of  $W$  is sampled i.i.d from  $N(0, \sigma^2/n)$ . Then we can consider the convex optimization problem in Lagrange form with a regularization parameter  $\lambda$ ,

$$L_\lambda(\hat{M}) = \frac{1}{2} \|A^\Omega - P_\Omega \mathcal{U}^* \hat{M}\|^2 + \sum_{k=1}^N \lambda \|\hat{M}(k)\|_*.$$

If we solve this problem via second order methods, it can become prohibitively expensive for large scale applications.

Another closely related formulation is the bi-convex formulation, which is to minimize the following objective function,

$$L_\lambda(X, Y) = \|A^\Omega - P_\Omega(X \star Y)\|^2 + \lambda(\|X\|^2 + \|Y\|^2).$$

This formulation is non-convex but it is computationally efficient in large-scale applications.

#### 5. Convolutional imputation

Now we propose a convolutional imputation algorithm that effectively finds the minimizer of the optimization problem

for a sequence of regularization parameters.

**Iterative imputation algorithm** The vanilla version of our imputation algorithm iteratively performs imputation of  $A^{\text{impute}} = P_{\Omega}(A) + P_{\Omega}^{\perp}(A^{\text{est}})$  and singular value soft-thresholding of  $\hat{A}^{\text{impute}}$  to solve the nuclear norm regularization problem. In the following, we denote singular value soft-thresholding as  $S_{\lambda_j}(\hat{A}) = P(\Sigma - \lambda_j I)_+ Q^*$ , where  $(\cdot)_+$  is the projection operator on the semi-definite cone, and  $\hat{A} = P\Sigma Q^*$  is the singular value decomposition.

---

**Algorithm 1** Iterative Imputation Algorithm
 

---

```

input :  $P_{\Omega}(A)$ .
    Initialization  $A_0^{\text{est}} = 0, t = 0$ .
    for  $\lambda_1 > \lambda_2 > \dots > \lambda_C$ , where  $\lambda_j = (\lambda_j(k)), k = 1, \dots, N$  do
        repeat
             $A^{\text{impute}} = P_{\Omega}(A) + P_{\Omega}^{\perp}(A_t^{\text{est}})$ .
             $\hat{A}^{\text{impute}} = \mathcal{U}A^{\text{impute}}$ .
             $\hat{A}_{t+1}^{\text{est}}(k) = S_{\lambda_j(k)}(\hat{A}^{\text{impute}}(k))$ .
             $A_{t+1}^{\text{est}} = \mathcal{U}^{-1}\hat{A}_{t+1}^{\text{est}}$ .
             $t = t + 1$ .
        until  $\|A_t^{\text{est}} - A_{t-1}^{\text{est}}\|^2 / \|A_{t-1}^{\text{est}}\|^2 < \epsilon$ .
        Assign  $A_{\lambda_j} = A_t^{\text{est}}$ .
    end for
output : the sequence of solutions  $A_{\lambda_1}, \dots, A_{\lambda_C}$ .
    
```

---

In the vanilla imputation algorithm, computing the full SVD on each iteration is very expensive for large matrices. For efficiency, we can use alternating ridge regression to compute reduced-rank SVD instead. Due to the limited space, we omit the detailed algorithm description here.

**Regularization path.** The sequence of regularization parameters is chosen such that  $\lambda_1(k) > \lambda_2(k) > \dots > \lambda_C(k)$  for each  $k$ . The solution for each iteration with  $\lambda_s$  is a warm start for the next iteration with  $\lambda_{s+1}$ . Our recommended choice is to choose  $\lambda_1(k)$  as  $\lambda_{\max}(\hat{A}^{\text{impute}}(k))$ , the largest singular value for  $\hat{A}^{\text{impute}}(k)$ , and decay  $\lambda_s$  at a constant speed  $\lambda_{s+1} = c\lambda_s$ .

**Computational complexity.** Now we analyze the computational complexity of the imputation algorithm. The cost of graph Fourier transform on matrix network is  $O(mnN^2)$ . When the graph is a periodic lattice, using fast Fourier transform(FFT), it is reduces to  $O(mnN \log N)$ . The cost of SVD is  $O(\min(mn^2, m^2n)N)$  for computing singular value soft-thresholding. Replacing SVD with alternating ridge regression reduces the complexity to  $O(r^2nN)$ . Therefore, the cost of each iteration is the sum of the cost of both parts, and the total cost would be that times total iteration steps.

**Convergence analysis of imputation algorithms** Now we show that the solution of our imputation algorithm converges asymptotically to a minimizer of the previously defined objective  $L_{\lambda}(\hat{M})$ . We generalize a single regularization parameter to a stack of regularization parameters  $\lambda = (\lambda(k))_{k=1}^N, \lambda(k) \geq 0$ .

Each step of our imputation algorithm is minimizing a surrogate  $Q_{\lambda}(\hat{M}|\hat{M}^{\text{old}})$  of the above objective function as

$$\|A^{\Omega} + P_{\Omega}^{\perp}\mathcal{U}^{-1}\hat{M}^{\text{old}} - \mathcal{U}^{-1}\hat{M}\|^2 + \sum_{k=1}^N \lambda(k)\|\hat{M}(k)\|_*.$$

The resulting minimizer forms a sequence  $\hat{M}_{\lambda}^t$  with starting point  $\hat{M}_{\lambda}^0$

$$\hat{M}_{\lambda}^{t+1} = \underset{\hat{M}}{\text{argmin}} \quad Q_{\lambda}(\hat{M}|\hat{M}_{\lambda}^t).$$

**Theorem 1.** *The imputation algorithm produces a sequence of iterates  $\hat{M}_{\lambda}^t$  that converges to the minimizer of  $L_{\lambda}(\hat{M})$ .*

The proof of this theorem is in the supplementary material. The main idea is to show that we show  $Q_{\lambda}$  decreases after every iteration and  $\hat{M}_{\lambda}^t$  is a Cauchy sequence, and the limit point is a stationary point of  $L_{\lambda}$ .

## 6. Exact recovery and phase transition

Let us now analyze the theoretical problem: what condition is needed for the non-uniform sampling  $\Omega$  and for the rank of  $\hat{A}$  such that our algorithm is guaranteed to perform accurate recovery with high probability? We focus on the noiseless case,  $\sigma = 0$ , and for simplicity of results, we assume  $m = n$ .

We first prove that one sufficient condition is that **average sampling rate**  $p = \frac{1}{N} \sum_{i=1}^N p_i = |\Omega|/(Nn^2)$  is greater than  $O(\frac{r}{n} \log^2(nN))$ . It is worth pointing out that the condition is only about the average sampling rate, therefore it includes the interesting case that a subset of matrices is completely unobserved.

### 6.1. Analysis on exact recovery guarantee

The optimization problem for recovery is

$$\begin{aligned} & \underset{\hat{M}}{\text{minimize}} \quad \sum_{k=1}^N \|\hat{M}(k)\|_*, \\ & \text{subject to} \quad A^{\Omega} = P_{\Omega}\mathcal{U}^* \hat{M}. \end{aligned}$$

We call the linear operator as the sensing operator  $\mathcal{R} = P_{\Omega}\mathcal{U}^*$ . If we vectorize  $\hat{M}$  and  $A^{\Omega}$  as  $\text{Vec}(\hat{M})$  and  $\text{Vec}(A^{\Omega})$ , then  $\mathcal{R}$  is a giant block diagonal matrix of size  $N^2n^4$ , each block is of size  $N^2$  and there are  $n^2$  such diagonal blocks. Each block is  $P_{ij}\mathcal{U}^*$ , where  $P_{ij}$  encodes the

sampling pattern for  $A_{ij}(k), k \in [1, N]$ . For the special case of node sampling,  $P_\Omega = P_G \otimes I$ , then the matrix form of  $\mathcal{R}$  is  $(P_G U^*) \otimes I$ , each block of it is the same partial unitary matrix  $P_G U^*$  of size  $N^2$ . To further simplify notation, we define

$$\|\hat{M}\|_{*,1} = \sum_{k=1}^N \|\hat{M}(k)\|_*.$$

Then the problem can be written as

$$\underset{\hat{M}}{\text{minimize}} \quad \|\hat{M}\|_{*,1}, \quad (1)$$

$$\text{subject to} \quad \mathcal{R}\hat{A} = \mathcal{R}\hat{M}. \quad (2)$$

The incoherence condition is a standard assumption for low-rank matrix recovery problems. Here we introduce an incoherence condition for the stack of matrices  $\hat{A}$  in spectral space, which is a weaker condition than the standard incoherence condition on each matrix  $\hat{A}(k)$ . This condition will be assumed to prove guarantee for recovery.

**Definition 6.1.** (*Incoherent condition.*) Let the SVD of  $\hat{A}(k)$  be  $\hat{A}(k) = V_1(k)E(k)V_2^*(k)$ . We now define an incoherent condition on the basis  $u_{kij} = u_k \otimes e_{ij}$  of the linear operator  $\mathcal{U}^*$ , where  $e_{ij}$  is the standard basis for  $n \times n$  matrix,  $u_k$  is the row vectors of inverse graph Fourier matrix  $U^*$ . We define that  $\hat{A}$  is incoherent with respect to  $\mathcal{U}^*$  with parameter  $\mu$  if

$$\frac{n}{rN} \max_{1 \leq i \leq n} \sum_{k=1}^N \|V_1^*(k)u_{ki1}\|^2 \leq \mu,$$

$$\frac{n}{rN} \max_{1 \leq j \leq n} \sum_{k=1}^N \|u_{k1j}V_2(k)\|^2 \leq \mu.$$

**Theorem 2.** We assume  $\hat{A}$  are low rank matrices that satisfy the incoherent condition with parameter  $\mu$ . There exist constants  $C, \gamma$ , if the average sampling rate  $p > C\mu_n^{\frac{r}{n}} \log^2(Nn)$ , then the solution to the optimization problem

$$\underset{\hat{M}}{\text{minimize}} \quad \|\hat{M}\|_{*,1},$$

$$\text{subject to} \quad \mathcal{R}\hat{A} = \mathcal{R}\hat{M}.$$

is unique and is exactly  $\hat{A}$  with probability  $1 - (Nn)^{-\gamma}$ .

The proof of this theorem is in the supplementary material. Here we sketch some intuition of the proof.

Geometrically, to prove the uniqueness is to show that the feasible set is tangent to the ball defined by the sum of nuclear norm at the point  $\hat{A}$ . Since the feasible set is a high dimensional plane, it is to show that there exists a normal vector  $K$  to a supporting hyperplane of the nuclear ball at  $\hat{A}$ , such that  $K$  is also normal to the high dimensional

plane of feasible set. This normal vector gives the intuition for a "dual certificate". Analytically, we will give a construction of "dual certificate" via a method called golf scheme. Overall, using incoherent condition and the existence of dual certificate, we will prove that for any nonzero  $\Delta = \hat{M} - \hat{A}$ , either  $\mathcal{R}\Delta \neq 0$ , or  $\|\hat{A} + \Delta\|_{*,1} > \|\hat{A}\|_{*,1}$ .

## 6.2. Numerical study of phase transition

To focus on the essential difficulty of the problem, we study the noiseless, node sampling setting: we generate a stack of low rank matrices in spectral space,  $\hat{A}(k) = (X^0(k))^T Y^0(k)$  for i.i.d Gaussian random matrix  $X^0(k), Y^0(k) \in \mathbf{R}^{r \times n}, k \in [1, N]$ . We also generate random graph  $G$  with different kinds of structure, which will be detailed below. Then we compute the matrix network  $A$  by inverse graph Fourier transform. And we observe a fraction of the matrices of  $A$  by node undersampling, where each matrix is either fully observed or fully missing.

We numerically investigate the phase transition for finite problem size  $(n, N)$ . In the following experiment, we fix the problem size  $(n, N)$  as  $(50, 100)$ , and vary the undersampling ratio  $p$  for node undersampling, and the rank  $r$  for  $\hat{A}(k)$ . For each set of parameters we generate low rank matrices  $\hat{A}$ , and observation  $A^\Omega$  according to our model. Then we solve the problem via the imputation algorithm to get solution  $\hat{M}$ . We measure the relative mean square error (rMSE)  $\|\hat{M} - \hat{A}\|/\|\hat{A}\|$ , and for multiple repeated experiments, we compute the success rate of exact recovery. (To tolerate numerical error, we define  $\text{rMSE} < 0.001$  as exact recovery.)

We can calculate the phase transition curve by fitting probit regressions using the success rate. For each  $p$ , we fit our observation of success/failure for different rank  $r$  with a probit curve, and find the transition point  $r(p)$  by solving  $\Phi(a_p r + b_p) = 1/2$ .

In Figure 5 and 6 we show results when the graph is a one-dimensional chain of length  $N$ . Figure 5 shows the logarithm of relative MSE while varying rank ratio  $r/n$  and undersampling ratio  $p$ . When  $r/n$  is large and  $p$  is small, the MSE is approximately equal to the undersampling ratio  $p$ , which means the optimization failed to recover the ground truth matrices. On the opposite side, when  $r/n$  is small and  $p$  is large, the MSE is very small, indicating we have successfully recovered the missing matrices. The transition between the two regions is very sharp. Figure 6 shows the probability of successful recovery repeating the experiments 5 times. It demonstrates the phase transition between total success and total failure.

We also experimented with other graph structures, and we find their phase transition curve overlapping each other, as shown in Figure 7. Here the linear graph refers to the pre-



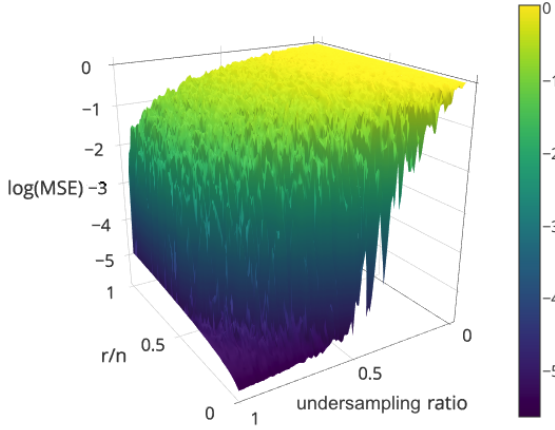


Figure 5. The MSE for different combination of undersampling ratio  $p$  and rank  $r/n$ . We observe that the transition between successful recovery ( $r\text{MSE} \approx 10^{-5}$ ) and failure ( $r\text{MSE} \approx p$ ) is very sharp.

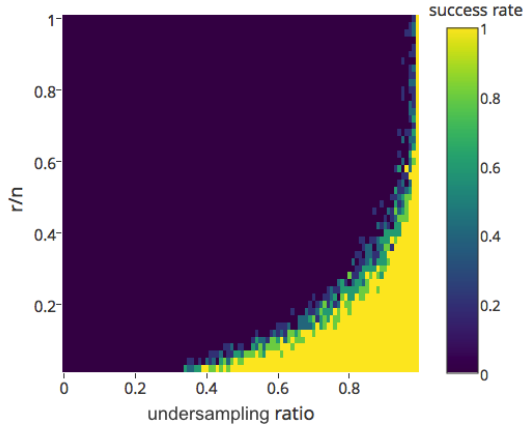


Figure 6. The phase transition graph with varying undersampling ratio and rank. We repeat the experiment 5 times for each parameter combination, and plot the success recovery rate ( $r\text{MSE} < 0.001$ )

vious case we have shown in Figure 5 and 6. Sparse graph refers to randomly generated graph with average degree  $0.05N$ . Dense graph refers to randomly generated graph with average degree  $0.5N$ . Unbalanced graph means 4 of the 100 graph nodes have average degree  $0.5N$ , while the other nodes have average degree  $0.05N$ . As we can see in Figure 7, the phase transition curves coincide for these different graphs. We call this phenomenon the **universality** of graph Fourier transform, which roots in the universality of unitary sensing matrix in traditional compressed sensing work.

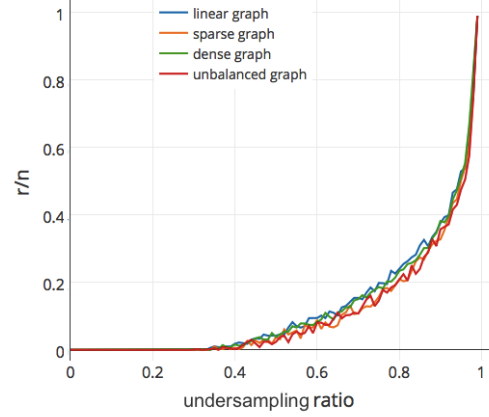


Figure 7. Phase transition curves for different types of graph structures. Explanation for each graph type is in the main text.

	$N_{\text{obs}} = 0.2N$ $p = 1$	$N_{\text{obs}} = 0.4N$ $p = 0.5$	$N_{\text{obs}} = 0.9N$ $p = 2/9$
$\sigma = 0$	0.116/0.000	0.100/0.038	0.088/0.061
$\sigma = 0.1$	0.363/0.495	0.348/0.411	0.339/0.365

Table 1. Average MSE of missing/(partially) observed matrices.

## 7. Experimental results

### 7.1. Synthetic feature matrices on Facebook network

We take the ego networks from the SNAP Facebook dataset (Leskovec & Mcauley, 2012). The combined ego networks form a connected graph with 4039 nodes and 88234 edges. All edges have equal weights. We construct synthetic feature matrices on each of the nodes by randomly generating  $X(k), Y(k) \in \mathbb{C}^{1 \times 50}$  in the spectral domain, and doing inverse graph Fourier transform to get  $A = \mathcal{U}^{-1}(X(k)Y(k))$ . The observation is generated by sampling  $N_{\text{obs}}$  matrices at sampling rate  $p$ , and adding i.i.d. Gaussian noise with mean 0 and variance  $\sigma^2/50$  to all observed entries. Here  $N_{\text{obs}} < N = 4039$  and the other matrices are completely unobserved.

We run our iterative imputation algorithm to recover  $A$  from this observation with varying parameters  $N_{\text{obs}}, p$ , and  $\sigma$ , and calculate the MSE between our estimation and the groundtruth. The results are summarized in Table 1. When there is no additive noise, we can recover all the matrices very well even with only 20% of entries observed across the matrix network. When there is additive noise, the MSE between reconstruction and groundtruth will grow proportionally.

## 7.2. MRI dataset

We use a normal cardiac MRI dataset called CETAU-TOMATIX from <http://www.osirix-viewer.com/datasets/>. The dataset has 88 frames with image dimension  $512 \times 416$ . This stack of MRI images scans through a human torso, and cardiovascular structures are highlighted. We corrupt the data by adding i.i.d. Gaussian noise with variance 0.1, and two frames are completely missing. The other 86 frames are sampled with 20% sampling rate using Bernoulli independent sampling.

We try to recover the image stack from the above observation, and compare our method with two baseline methods, the nuclear norm minimization method for matrix completion on each frame, and the tensor completion with trace norm regularization method (Tomioka et al., 2010). The implementation of tensor completion from the original paper converges very slowly, and cannot recover frames that are unobserved. In Figure 1 we show reconstructed images from our method and matrix completion method for three frames out of the 88 corrupted frames, where the first and third frames are observed with sampling rate  $p = 20\%$ , the second frame is not observed. Compared to matrix completion, our convolutional imputation is able to reconstruct the unobserved frame. In addition, our reconstruction for partially observed frames are visually smoother and recovers more detailed information. A quantitative comparison is given in Figure 8. The relative MSE between reconstruction and groundtruth is significantly smaller from our imputation algorithm than the baseline.

## 8. Conclusion

In this work, we introduce the statistical model of matrix networks. A matrix network has a set of matrices of the same size, and they are connected by a weighted undirected graph. Based on the model, we consider the problem of matrix network completion from incomplete observations. The underlying network enables us to handle cases when observation on each matrix is highly unbalanced, especially when some of the matrices are completely unobserved. We propose the structural assumption that a matrix network can be approximated by the graph convolution of two sets of low rank matrices living on the same graph. Under this assumption we formulate an optimization problem and proposed an iterative imputation algorithm that recovers the matrix networks accurately. We prove exact recovery guarantee for the optimization problem when the average sampling rate is greater than  $O(\frac{\epsilon}{n} \log^2(nN))$ . Furthermore, we numerically characterize the success regime for varying rank and sampling rate, and discover a new phase transition phenomenon that is universal in different graph structures. We demonstrate the algorithm in completing MRI scans and on a graph from Facebook user network.

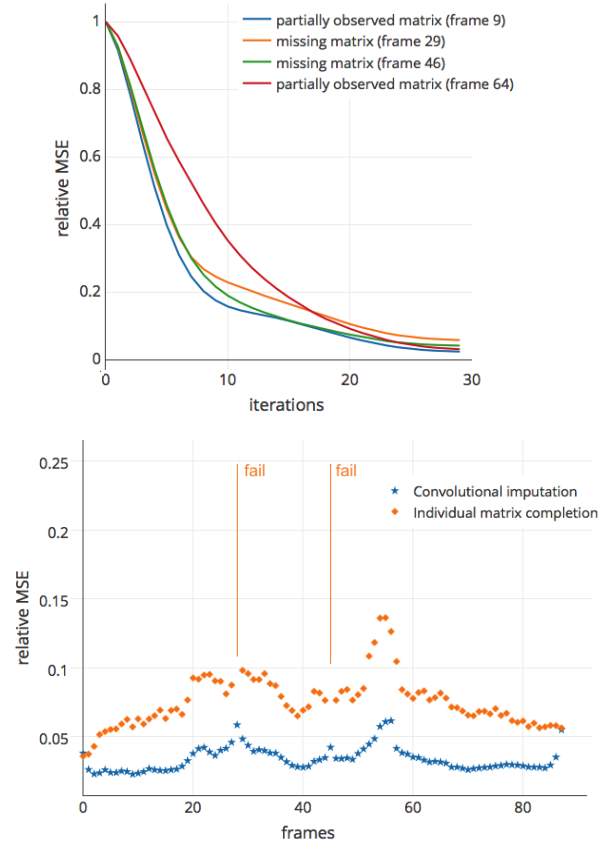


Figure 8. Upper: Convergence of imputation algorithm. The relative MSE of two partially observed frames and two missed frames are shown. Lower: Comparison of relative MSE for all frames, matrix completion fails at the missed frames 29 and 46, and is dominated by convolutional completion for other frames.

## References

- Bruna, Joan, Zaremba, Wojciech, Szlam, Arthur, and LeCun, Yann. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.
- Cai, Jian-Feng, Candès, Emmanuel J, and Shen, Zuowei. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- Candès, Emmanuel and Recht, Benjamin. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.
- Candès, Emmanuel J and Plan, Yaniv. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.
- Candès, Emmanuel J and Tao, Terence. The power of convex relaxation: Near-optimal matrix completion. *IEEE*



- Transactions on Information Theory*, 56(5):2053–2080, 2010.
- Donoho, David, Gavish, Matan, et al. Minimax risk of matrix denoising by singular value thresholding. *The Annals of Statistics*, 42(6):2413–2440, 2014.
- Filipović, Marko and Jukić, Ante. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional systems and signal processing*, 26(3):677–692, 2015.
- Gandy, Silvia, Recht, Benjamin, and Yamada, Isao. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.
- Gavish, Matan and Donoho, David L. The optimal hard threshold for singular values is  $4/\sqrt{3}$ . *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.
- Gross, David. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.
- Hastie, T and Mazumder, R. softimpute: Matrix completion via iterative soft-thresholded svd. *R package version*, 1, 2015.
- Hastie, Trevor, Mazumder, Rahul, Lee, Jason D, and Zadeh, Reza. Matrix completion and low-rank svd via fast alternating least squares. *J. Mach. Learn. Res*, 16(1):3367–3402, 2015.
- Henaff, Mikael, Bruna, Joan, and LeCun, Yann. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.
- Kernfeld, Eric, Kilmer, Misha, and Aeron, Shuchin. Tensor–tensor products with invertible linear transforms. *Linear Algebra and its Applications*, 485:545–570, 2015.
- Keshavan, Raghunandan H, Montanari, Andrea, and Oh, Sewoong. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010a.
- Keshavan, Raghunandan H, Montanari, Andrea, and Oh, Sewoong. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010b.
- Leskovec, Jure and McAuley, Julian J. Learning to discover social circles in ego networks. pp. 539–547, 2012.
- Liu, Ji, Musialski, Przemyslaw, Wonka, Peter, and Ye, Jieping. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.
- Liu, Xiao-Yang, Aeron, Shuchin, Aggarwal, Vaneet, Wang, Xiaodong, and Wu, Min-You. Adaptive sampling of rf fingerprints for fine-grained indoor localization. *IEEE Transactions on Mobile Computing*, 15(10):2411–2423, 2016.
- Mazumder, Rahul, Hastie, Trevor, and Tibshirani, Robert. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.
- Monajemi, Hatef and Donoho, David L. Sparsity/undersampling tradeoffs in anisotropic undersampling, with applications in mr imaging/spectroscopy. *arXiv preprint arXiv:1702.03062*, 2017.
- Tomioka, Ryota, Hayashi, Kohei, and Kashima, Hisashi. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.
- Zhang, Zemin and Aeron, Shuchin. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2016.